Creating Disjunctive Logical Forms from Aligned Sentences for Grammar-Based Paraphrase Generation

Scott Martin and **Michael White**
Department of Linguistics
The Ohio State University
Columbus, Ohio, USA
{scott,mwhite}@ling.ohio-state.edu

Abstract

We present a method of creating disjunctive logical forms (DLFs) from aligned sentences for grammar-based paraphrase generation using the OpenCCG broad coverage surface realizer. The method takes as input word-level alignments of two sentences that are paraphrases and projects these alignments onto the logical forms that result from automatically parsing these sentences. The projected alignments are then converted into phrasal edits for producing DLFs in both directions, where the disjunctions represent alternative choices at the level of semantic dependencies. The resulting DLFs are fed into the OpenCCG realizer for n-best realization, using a pruning strategy that encourages lexical diversity. After merging, the approach yields an n-best list of paraphrases that contain grammatical alternatives to each original sentence, as well as paraphrases that mix and match content from the pair. A preliminary error analysis suggests that the approach could benefit from taking the word order in the original sentences into account. We conclude with a discussion of plans for future work, highlighting the method's potential use in enhancing automatic MT evaluation.

1 Introduction

In this paper, we present our initial steps towards merging the grammar-based and data-driven paraphrasing traditions, highlighting the potential of our approach to enhance the automatic evaluation of machine translation (MT). Kauchak and Barzilay (2006) have shown that creating synthetic reference sentences by substituting synonyms from Wordnet into the original reference sentences can increase the number of exact word matches with an MT system's output and yield significant improvements in correlations of BLEU (Papineni et al., 2002) scores with human judgments of translation adequacy. Madnani (2010) has also shown that statistical machine translation technique can be employed in a monolingual setting, together with paraphrases acquired using Bannard and Callison-Burch's (2005) pivot method, in order to enhance the tuning phase of training an MT system by augmenting a reference translation with automatic paraphrases. Earlier, Barzilay and Lee (2003) and Pang et al. (2003) developed approaches to aligning multiple reference translations in order to extract paraphrases and generate new sentences. By starting with reference sentences from multiple human translators, these data-driven methods are able to capture subtle, highly-context sensitive word and phrase alternatives. However, the methods are not particularly adept at capturing variation in word order or the use of function words that follow from general principles of grammar. By contrast, grammar-based paraphrasing methods in the natural language generation tradition (Iordanskaja et al., 1991; Elhadad et al., 1997; Langkilde and Knight, 1998; Stede, 1999; Langkilde-Geary, 2002; Velldal et al., 2004; Gardent and Kow, 2005; Hogan et al., 2008) have the potential to produce many such grammatical alternatives: in particular, by parsing a reference sentence to a representation that can be used as the input to a surface realizer, grammar-based paraphrases can be generated if the realizer supports n-best output. To our knowledge though, methods of using a grammar-based surface realizer together with multiple aligned reference sentences to produce synthetic

Workshop on Monolingual Text-To-Text Generation, pages 74–83,

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 74–83,
Portland, Oregon, 24 June 2011. ©2011 Association for Computational Linguistics

Source	Liu Lefei says that [in the long *term*] , in terms of *asset* allocation, overseas investment should occupy a certain *proportion* of [an insurance company's overall allocation] .
Reference	Liu Lefei said that in terms of *capital* allocation , outbound investment should make up a certain *ratio* of [overall allocations for insurance companies] [in the long *run*] .
Paraphrase	Liu Lefei says that [in the long *run*], in terms of *capital* allocation, overseas investment should occupy *the* certain *ratio* of an [insurance company's overall allocation]

Table 1: Zhao et al.'s (2009) similarity example, with italics added to show word-level substitutions, and square brackets added to show phrase location or construction mismatches. Here, the source sentence (itself a reference translation) has been paraphrased to be more like the reference sentence.

references have not been investigated.[1]

As an illustration of the need to combine grammatical paraphrasing with data-driven paraphrasing, consider the example that Zhao et al. (2009) use to illustrate the application of their paraphrasing method to similarity detection, shown in Table 1. Zhao et al. make use of a large paraphrase table, similar to the phrase tables used in statistical MT, in order to construct paraphrase candidates. (Like thesauri or WordNet, such resources are complementary to the ones we make use of here.) To test their system's ability to paraphrase reference sentences in service of MT evaluation, they attempt to paraphrase one reference translation to make it more similar to another reference translation; thus, in Table 1, the source sentence (itself a reference translation) has been paraphrased to be more like the (other) reference sentence. As indicated by italics, their system has successfully paraphrased *term*, *asset* and *proportion* as *run*, *capital* and *ratio*, respectively (though *the certain* seems to have been mistakenly substituted for *a certain*). However, their system is not capable of generating a paraphrase with *in the long run* at the end of the sentence, nor can it rephrase *insurance company's overall allocation* as *overall allocations for insurance companies*, which would seem to require access to more general grammatical knowledge.

To combine grammar-based paraphrasing with lexical and phrasal alternatives gleaned from multiple reference sentences, our approach takes advantage of the OpenCCG realizer's ability to generate from **disjunctive logical forms** (DLFs), i.e. packed semantic dependency graphs (White, 2004; White, 2006a; White, 2006b; Nakatsu and White, 2006; Espinosa et al., 2008; White and Rajkumar, 2009). In principle, semantic dependency graphs offer a better starting point for paraphrasing than the syntax trees employed by Pang et. al, as paraphrases can generally be expected to be more similar at the level of unordered semantic dependencies than at the level of syntax trees. Our method starts with word-level alignments of two sentences that are paraphrases, since the approach can be used with any alignment method from the MT (Och and Ney, 2003; Haghighi et al., 2009, for example) or textual inference (MacCartney et al., 2008, inter alia) literature in principle. The alignments are projected onto the logical forms that result from automatically parsing these sentences. The projected alignments are then converted into phrasal edits for producing DLFs in both directions, where the disjunctions represent alternative choices at the level of semantic dependencies. The resulting DLFs are fed into the OpenCCG realizer for n-best realization. In order to enhance the variety of word and phrase choices in the n-best lists, a pruning strategy is used that encourages lexical diversity. After merging, the approach yields an n-best list of paraphrases that contain grammatical alternatives to each original sentence, as well as paraphrases that mix and match content from the pair.

The rest of the paper is organized as follows. Section 2 provides background on surface realization with OpenCCG and DLFs. Section 3 describes our

[1]The task is not unrelated to sentence fusion in multidocument summarization (Barzilay and McKeown, 2005), except there the goal is to produce a single, shorter sentence from multiple related input sentences.

method of creating DLFs from aligned paraphrases. Finally, Section 4 characterizes the recurring errors and concludes with a discussion of related and future work.

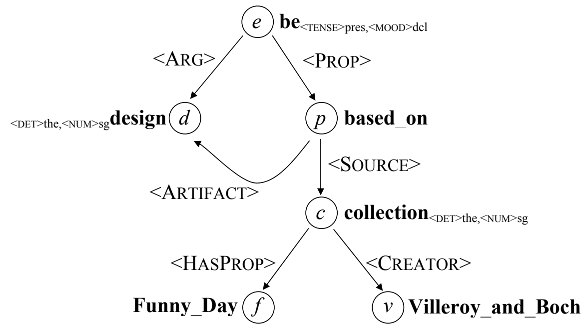
2 Surface Realization with OpenCCG

OpenCCG is an open source Java library for parsing and realization using Baldridge's multimodal extensions to CCG (Steedman, 2000; Baldridge, 2002). In the chart realization tradition (Kay, 1996), the OpenCCG realizer takes logical forms as input and produces strings by combining signs for lexical items. Alternative realizations are scored using integrated n-gram and perceptron models (White and Rajkumar, 2009), where the latter includes syntactic features from Clark and Curran's (2007) normal form model as well as discriminative n-gram features (Roark et al., 2004). Hypertagging (Espinosa et al., 2008), or supertagging for surface realization, makes it practical to work with broad coverage grammars. For parsing, an implementation of Hockenmaier and Steedman's (2002) generative model is used to select the best parse. The grammar is automatically extracted from a version of the CCGbank (Hockenmaier and Steedman, 2007) with Propbank (Palmer et al., 2005) roles projected onto it (Boxwell and White, 2008).

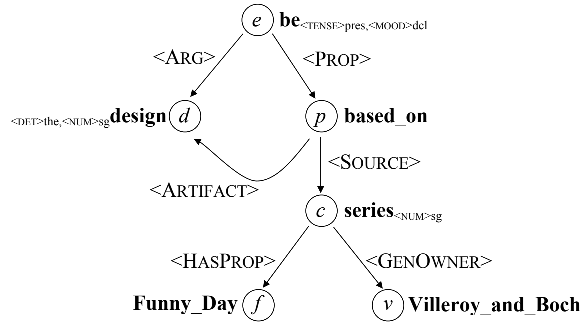
A distinctive feature of OpenCCG is the ability to generate from disjunctive logical forms (White, 2006a). This capability has many benefits, such as enabling the selection of realizations according to predicted synthesis quality (Nakatsu and White, 2006), and avoiding repetition in the output of a dialogue system (Foster and White, 2007). Disjunctive inputs make it possible to exert fine-grained control over the specified paraphrase space. In the chart realization tradition, previous work has not generally supported disjunctive logical forms, with Shemtov's (Shemtov, 1997) more complex approach as the only published exception.

An example disjunctive input from the COMIC system appears in Figure 1(c).[2] Semantic dependency graphs such as these—represented internally in Hybrid Logic Dependency Semantics

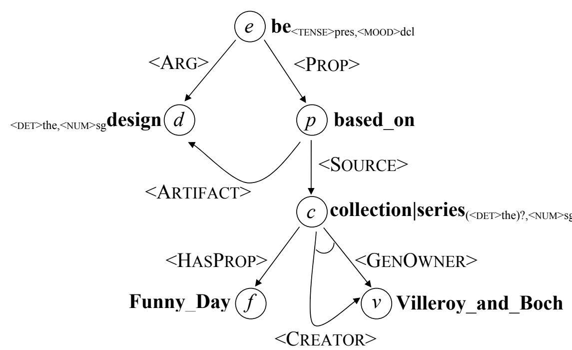
[2]To simplify the exposition, the features specifying information structure and deictic gestures have been omitted, as have the semantic sorts of the discourse referents.



(a) Semantic dependency graph for *The design (is|'s) based on the Funny Day collection by Villeroy and Boch.*



(b) Semantic dependency graph for *The design (is|'s) based on Villeroy and Boch's Funny Day series.*



(c) Disjunctive semantic dependency graph covering (a)-(b), i.e. *The design (is|'s) based on (the Funny Day (collection|series) by Villeroy and Boch | Villeroy and Boch's Funny Day (collection|series)).*

Figure 1: Two similar logical forms from the COMIC system as semantic dependency graphs, together with a disjunctive logical form representing their combination as a packed semantic dependency graph.

(Baldridge and Kruijff, 2002; White, 2006b), or HLDS—constitute the input to the OpenCCG realizer.[3] This graph allows a free choice between the domain synonyms *collection* and *series*, as indicated by the vertical bar between their respective predications. The graph also allows a free choice between the ⟨CREATOR⟩ and ⟨GENOWNER⟩ relations—lexicalized via *by* and the possessive, respectively—connecting the head c (*collection* or *series*) with the dependent v (for *Villeroy and Boch*); this choice is indicated by an arc between the two dependency relations. Finally, the determiner feature (⟨DET⟩the) on c is indicated as optional, via the question mark. Note that as an alternative, the determiner feature could have been included in the disjunction with the ⟨CREATOR⟩ relation (though this would have been harder to show graphically); however, it is not necessary to do so, as constraints in the lexicalized grammar will ensure that the determiner is not generated together with the possessive.

3 Constructing DLFs from Aligned Paraphrases

To develop our approach, we use the gold-standard alignments in Cohn et al.'s (2008) paraphrase corpus. This corpus is constructed from three monolingual sentence-aligned paraphrase subcorpora from differing text genres, with word-level alignments provided by two human annotators. We parse each corpus sentence pair using the OpenCCG parser to yield a logical form (LF) as a semantic dependency graph with the gold-standard alignments projected onto the LF pair. Disjunctive LFs are then constructed by inspecting the graph structure of each LF in comparison with the other. Here, an alignment is represented simply as a pair $\langle n1, n2\rangle$ where $n1$ is a node in the first LF and $n2$ a node in the second LF. As Cohn et al.'s corpus contains some **block alignments**, there are cases where a single node is aligned to multiple nodes in the other sentence of the paraphrase.

A semantic dependency is represented as graph $\mathcal{G} = \langle N, E\rangle$, where $N = \mathsf{nodes}(\mathcal{G})$ is the set of nodes in $\mathcal{G}$ and $E = \mathsf{edges}(\mathcal{G})$ is the set of edges in $\mathcal{G}$. An edge e is a labeled dependency between nodes, with $\mathsf{source}(e)$ denoting the source node, $\mathsf{target}(e)$ the target node, and $\mathsf{label}(e)$ the relation e represents. For $n, n' \in \mathsf{nodes}(\mathcal{G})$ members of the set of nodes for some graph $\mathcal{G}$, $n' \in \mathsf{parents}(n)$ if and only if there is an edge $e \in \mathsf{edges}(\mathcal{G})$ with $n' = \mathsf{source}(e)$ and $n = \mathsf{target}(e)$. The set $\mathsf{ancestors}(n)$ models the transitive closure of the 'parent-of' relation: $a \in \mathsf{ancestors}(n)$ if and only if there is some $p \in \mathsf{parents}(n)$ such that either $a = p$ or $a \in \mathsf{ancestors}(p)$. Nodes in a graph additionally bear associated predicates and semantic features that are derived during the parsing process.

3.1 The Algorithm

As a preprocessing step, we first characterize the difference between two LFs as a set of edit operations via MAKEEDITS$(g1, g2, alignments)$, as detailed in Algorithm 1. An **insert** results when the second graph contains an unaligned subgraph. Similarly, an unaligned subgraph in the first LF is characterized by a **delete** operation. For both inserts and deletes, only the head of the inserted or deleted subgraph is represented as an edit in order to reflect the fact that these operations can encompass entire subgraphs. A **substitution** occurs when a subgraph in the first LF is aligned to one or more subgraphs in the second LF. The case where subgraphs are block aligned corresponds to a multi-word phrasal substitution (for example, the substitution of *Goldman* for *The US investment bank* in paraphrase (2), below). The DLF generation process is then driven by these edit operations.

DLFs are created for each sentence by DISJUNCTIVIZE$(g1, g2, alignments)$ and DISJUNCTIVIZE$(g2, g1, alignments)$, respectively, where $g1$ is the first sentence's LF and $g2$ the LF of the second (see Algorithm 2). The DLF construction process takes as inputs a pair of dependency graphs $\langle g1, g2\rangle$ and a set of word-level alignments from Cohn et al.'s (2008) paraphrase corpus projected onto the graphs. This process creates a DLF by merging or making optional material from the sec-

[3] To be precise, the HLDS logical forms are descriptions of semantic dependency graphs, which in turn can be interpreted model theoretically via translation to Discourse Representation Theory (Kamp and Reyle, 1993), as White (2006b) explains. A disjunctive logical form is thus a description of a set of semantic dependency graphs. (As the LFs derived using CCGbank grammars do not represent quantifier scope properly, it would be more accurate to call them quasi-LFs; as this issue does not appear to impact the realization or DLF creation algorithms, however, we have employed the simpler term.)

Algorithm 1 Preprocesses a pair of aligned LFs representing a paraphrase into edit operations.

procedure $\textsc{MakeEdits}(g1, g2, alignments)$
  for all $i \in \{n \in \mathsf{nodes}(g2) \mid \neg\exists x.\langle x, n\rangle \in alignments\}$ **do** $\triangleright$ inserts
    if $\neg\exists p.p \in \mathsf{parents}(i) \wedge \neg\exists x.\langle x, p\rangle \in alignments$ **then**
      $\mathsf{insert}(i)$
  for all $d \in \{n \in \mathsf{nodes}(g1) \mid \neg\exists y.\langle n, y\rangle \in alignments\}$ **do** $\triangleright$ deletes
    if $\neg\exists p.p \in \mathsf{parents}(d) \wedge \neg\exists y.\langle p, y\rangle \in alignments$ **then**
      $\mathsf{delete}(d)$
  for all $s \in \mathsf{nodes}(g1)$ **do** $\triangleright$ substitutions
    if $\exists y.\langle s, y\rangle \in alignments \wedge \neg\exists z.z \in \mathsf{parents}(y) \wedge \langle s, z\rangle \in alignments$ **then**
      $\mathsf{substitution}(s, y)$

Algorithm 2 Constructs a disjunctive LF from an aligned paraphrase.

procedure $\textsc{Disjunctivize}(g1, g2, alignments)$
  $\textsc{MakeEdits}(g1, g2, alignments)$
  for all $i \in \{n \in \mathsf{nodes}(g2) \mid \mathsf{insert}(n)\}$ **do**
    for all $p \in \{e \in \mathsf{edges}(g2) \mid i = \mathsf{target}(e)\}$ **do**
      for all $\langle n1, n2\rangle \in \{\langle x, y\rangle \in alignments \mid y = \mathsf{source}(p)\}$ **do**
        $\mathsf{option}(n1, p)$
  for all $d \in \{n \in \mathsf{nodes}(g1) \mid \mathsf{delete}(n)\}$ **do**
    for all $p \in \{e \in \mathsf{edges}(g1) \mid d = \mathsf{target}(e)\}$ **do**
      $\mathsf{option}(\mathsf{source}(p), p)$
  for all $s \in \{n \in \mathsf{nodes}(g1) \mid \exists y.\mathsf{substitution}(n, y)\}$ **do**
    for all $p \in \mathsf{parents}(s)$ **do**
      $\mathsf{choice}(p, \{e \in \mathsf{edges}(g2) \mid \mathsf{substitution}(s, \mathsf{target}(e)) \wedge \langle p, \mathsf{source}(e)\rangle \in alignments\})$

ond LF into the first LF.

As Algorithm 2 describes, first the inserts (line 3) and deletes (line 7) are handled. In the case of inserts, for each node i in the second LF that is the head of an inserted subgraph, we find every $n2$ that is the source of an edge p whose target is i. The edge p is added as an option for each node $n1$ in the first LF that is aligned to $n2$. The process for deletes is similar, modulo direction reversal. We find every edge p whose target is d, where d is the head of an unaligned subgraph in the first sentence, and make p an option for the parent node $\mathsf{source}(p)$. With both inserts and deletes, the intuitive idea is that an unaligned subgraph should be treated as an optional dependency from its parent.

The following corpus sentence pair demonstrates the handling of inserts/deletes:

(1) a. Justices said that the constitution allows the government to administer drugs only in limited circumstances.
b. In a 6-3 ruling, the justices said such anti-psychotic drugs can be used only in limited circumstances.

In the DLF constructed for (1a), the node representing the word *drugs* has two alternate children that are not present in the first sentence itself (i.e., are inserted), *such* and *anti-psychotic*, both of which are in the modifier relation to *drugs*. This happens because *drugs* is aligned to the word *drugs* in (1b), which has the modifier child nodes. The second sentence also contains the insertion *In a 6-3 ruling*. This entire subgraph is represented as an optional modifier of *said*. Finally, the determiner *the* is inserted before *justices* in the second sentence. This determiner is also represented as an optional edge from *justices*. Figure 2 shows the portion of the DLF reflecting the optional modifier *In a 6-3 ruling* and optional determiner *the*.

For substitutions (line 10), we consider each subgraph-heading node s in the first LF that is substituted for some node y in the second LF that is also a subgraph head. Then for each parent p of s, the choices for p are contained in the set of edges whose source is aligned to p and whose target is a substitution for s. The intuition is that for each node p in the first LF with an aligned subgraph c, there is a disjunction between c and the child subgraphs of the

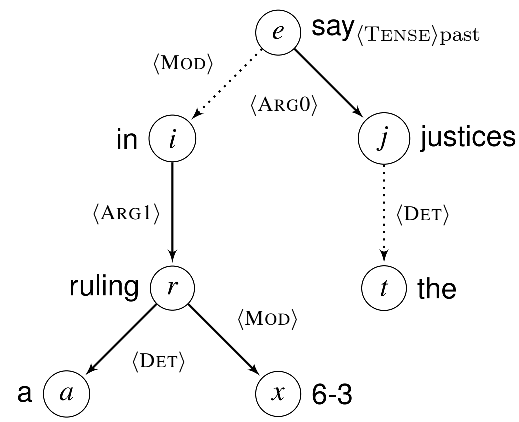


Figure 2: Disjunctive LF subgraph for the alternation *(In a 6-3 ruling)*? *(the)*? *justices said . . .* in paraphrase (1). The dotted lines represent optional edges, and some semantic features are suppressed for readability.

node that p is aligned to in the second LF. For efficiency, in the special case of substitutions involving single nodes rather than entire subgraphs, only the semantic predicates are disjoined.[4]

To demonstrate, consider the following corpus sentence pair involving a phrasal substitution:

(2) a. The US investment bank said: we believe the long-term prospects for the energy sector in the UK remain attractive.
b. We believe the long-term prospects for the energy sector in the UK remain attractive, Goldman said.

In this paraphrase, the subtree *The US investment bank* in (2a) is aligned to the single word *Goldman* in (2b), but their predicates are obviously different. The constructed DLF contains a choice between *Goldman* and *The US investment bank* as the subject of *said*. Figure 3 illustrates the relevant subgraph of the DLF constructed from the *Goldman* paraphrase with a choice between subjects (⟨ARG0⟩). This disjunction arises because *said* in the first sentence is aligned to *said* in the second, and *The US investment bank* is the subject of *said* in the first while *Goldman* is its subject in the second. Note that, since the substitution is a phrasal (block-aligned) one, the constructed DLF forces a choice between *Goldman* and the entire subgraph headed by *bank*, not between

[4]We leave certain more complex cases, e.g. multiple nodes with aligned children, for future work.

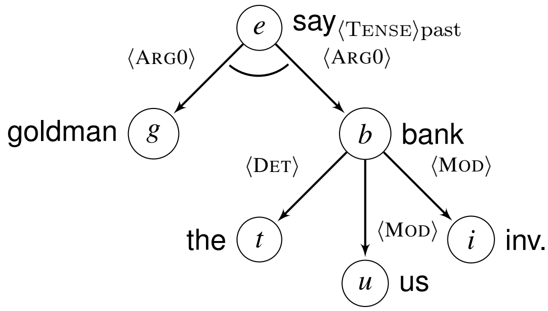


Figure 3: Disjunctive LF subgraph for the alternation (*Goldman* | *The US investment bank*) *said* ... in paraphrase (2). The arc represents the two choice edges for the ⟨ARG0⟩ relation from *say*. Certain semantic dependencies are omitted, and the word *investment* is abbreviated to save space.

Goldman and each of *bank*'s dependents (*the, US*, and *investment*).

4 Discussion and Future Work

With a broad coverage grammar, we have found that most of the realization alternatives in an n-best list tend to reuse the same lexical choices, with the differences mostly consisting of alternate word orders or use of function words or punctuation. Accordingly, in order to enhance the variety of word and phrase choices in the n-best lists, we have taken advantage of the API-level support for plugging in custom pruning strategies and developed a custom strategy that encourages lexical diversity. This strategy groups realizations that share the same open class stems into equivalence classes, where new equivalence classes are favored over new alternatives within the same equivalence class in filling up the n-best list.

Using this lexical diversity pruning strategy, an example of the paraphrases generated after DLF creation appears in Table 2. In the example, *the girl* and *brianna* are successfully alternated, as are *her mother's* and *the (bedroom)*. The example also includes a reasonable Heavy-NP shift, with *into the bedroom* appearing before the NP list. Without the lexical diversity pruning strategy, the phrase *her mother's* does not find its way into the n-best list. The paraphrases also include a mistaken change in tense from *had* to *has* and a mysterious inclusion of *including*. Interestingly though, these mistakes follow in the n-best list alternatives that are otherwise the same, suggesting that a final pruning of the list may make it possible to keep only generally good paraphrases. (Note that the appositive *33* in the second reference sentence also has been dropped, most likely since the pruning strategy does not include numbers in the set of content words at present.)

Although we have not yet formally evaluated the paraphrases, we can already characterize some recurring errors. Named entities are an issue since we have not incorporated a named entity recognizer; thus, the realizer is apt to generate *O. Charles Prince* instead of *Charles O. Prince*, for example. Worse, *medical examiner 's spokeswoman ellen borakove* is realized both correctly and as *medical examiner 's ellen spokeswoman borakove*. Naturally, there are also paraphrasing errors that stem from parser errors. Certainly with named entities, though perhaps also with parser errors, we plan to investigate whether we can take advantage of the word order in the reference sentence in order to reduce the number of mistakes. Here, we plan to investigate whether a feature measuring similarity in word order to the original can be balanced against the averaged perceptron model score in a way that allows new paraphrases to be generated while sticking to the original order in cases of uncertainty. Initial experiments with adding to the perceptron model score an n-gram precision score (approximating BLEU) with an appropriate weight indicate that realizations including the correct word order in names such as *Charles O. Prince* can be pushed to the top of the n-best list, though it remains to be verified that the weight for the similarity score can be adequately tuned with held-out data. Incorporating a measure of similarity to the original reference sentences into realization ranking is a form of what Madnani (2010) calls a **self-paraphrase bias**, though a different one than his method of adjusting the probability mass assigned to the original.

In future work, we plan to evaluate the generated paraphrases both intrinsically and extrinsically in combination with MT evaluation metrics. With the intrinsic evaluation, we expect to examine the impact of parser and alignment errors on the paraphrases, and the extent to which these can be mitigated by a self-paraphrase bias, along with the impact of the lexical diversity pruning strategy on the

Reference 1	lee said brianna had dragged food , toys and other things into the bedroom .
Realizations	lee said *the girl* had dragged food , toys and other things into the bedroom .
	lee said brianna had dragged food , toys and other things into the bedroom .
	lee said , the girl had dragged [into the bedroom] food , toys and other things .
	lee said the girl has dragged into the bedroom food , toys and other things .
	lee said , brianna had dragged into the bedroom food , toys and other things .
	lee said the girl had dragged food , toys and other things into *her mother 's* bedroom .
	lee said , the girl had dragged into her mother 's bedroom food , toys and other things .
	lee said brianna had dragged food , toys and other things into her mother 's bedroom .
	lee said the girl had dragged food , toys and other things into including the bedroom .
	lee said , brianna had dragged into her mother 's bedroom food , toys and other things .
Reference 2	lee , 33 , said the girl had dragged the food , toys and other things into her mother 's bedroom .
Realizations	lee said the girl had dragged [into *the* bedroom] the food , toys and other things .
	lee said , the girl had dragged into the bedroom the food , toys and other things .
	lee said the girl has dragged into the bedroom the food , toys and other things .
	lee said *brianna* had dragged the food , toys and other things into the bedroom .
	lee said , brianna had dragged into the bedroom the food , toys and other things .
	lee said the girl had dragged the food , toys and other things into her mother 's bedroom .
	lee said brianna had dragged into her mother 's bedroom the food , toys and other things .
	lee said , the girl had dragged into her mother 's bedroom the food , toys and other things .
	lee said brianna had dragged the food , toys and other things into her mother 's bedroom .
	lee said the girl had dragged the food , toys and other things into including the bedroom .

Table 2: Example n-best realizations starting from each reference sentence. Alternative phrasings from the other member of the pair are shown in italics the first time, and alternative phrase locations are shown in square brackets. Mistakes are underlined, and suppressed after the first occurrence in the list.

number of acceptable paraphrases in the n-best list.

With the extrinsic evaluation, we plan to investigate whether n-best paraphrase generation using the methods described here can be used to augment a set of reference translations in such a way as to increase the correlation of automatic metrics with human judgments. As Madnani observes, generated paraphrases of reference translations may be either untargeted or targeted to specific MT hypotheses. In the case of targeted paraphrases, the generated paraphrases then approximate the process by which automatic translations are evaluated using HTER (Snover et al., 2006), with a human in the loop, as the closest acceptable paraphrase of a reference sentence should correspond to the version of the MT hypothesis with minimal changes to make it acceptable. While in principle we might similarly acquire paraphrase rules using the pivot method, as in Madnani's approach, such rules would be quite noisy, as it is a difficult problem to characterize the contexts in which words or phrases can be acceptably substituted. Thus, our immediate focus will be on generating synthetic references with high precision, relying on grammatical alternations plus contextually acceptable alternatives present in multiple reference translations, given that metrics such as METEOR (Banerjee and Lavie, 2005) and TERp (Snover et al., 2010) can now employ paraphrase matching as part of their scoring, complementing what can be done with our methods. To the extent that we can maintain high precision in generating synthetic reference sentences, we may expect the correlations between automatic metric scores and human judgments to improve as the task of the metrics becomes simpler.

Acknowledgements

This work was supported in part by NSF grant number IIS-0812297. We are also grateful to Trevor Cohn for help with the paraphrase data.

References

Jason Baldridge and Geert-Jan Kruijff. 2002. Coupling CCG and Hybrid Logic Dependency Semantics. In *Proc. ACL-02*.

Jason Baldridge. 2002. *Lexically Specified Derivational*

Control in Combinatory Categorial Grammar. Ph.D. thesis, School of Informatics, University of Edinburgh.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proc. of the ACL-05 Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*.

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proc. ACL-05*, pages 597–604.

Regina Barzilay and Lillian Lee. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *Proc. of NAACL-HLT*.

Regina Barzilay and Kathleen McKeown. 2005. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–327.

Stephen Boxwell and Michael White. 2008. Projecting Propbank roles onto the CCGbank. In *Proc. LREC-08*.

Stephen Clark and James R. Curran. 2007. Wide-Coverage Efficient Statistical Parsing with CCG and Log-Linear Models. *Computational Linguistics*, 33(4):493–552.

Trevor Cohn, Chris Callison-Burch, and Mirella Lapata. 2008. Constructing corpora for the development and evaluation of paraphrase systems. *Computational Linguistics*, 34(4):597–614.

M. Elhadad, J. Robin, and K. McKeown. 1997. Floating constraints in lexical choice. *Computational Linguistics*, 23(2):195–239.

Dominic Espinosa, Michael White, and Dennis Mehay. 2008. Hypertagging: Supertagging for surface realization with CCG. In *Proceedings of ACL-08: HLT*, pages 183–191, Columbus, Ohio, June. Association for Computational Linguistics.

Mary Ellen Foster and Michael White. 2007. Avoiding repetition in generated text. In *Proceedings of the 11th European Workshop on Natural Language Generation (ENLG 2007)*.

Claire Gardent and Eric Kow. 2005. Generating and selecting grammatical paraphrases. In *Proc. ENLG-05*.

Aria Haghighi, John Blitzer, John DeNero, and Dan Klein. 2009. Better word alignments with supervised ITG models. In *Proceedings of ACL*, pages 923–931, Suntec, Singapore, August. Association for Computational Linguistics.

Julia Hockenmaier and Mark Steedman. 2002. Generative models for statistical parsing with Combinatory Categorial Grammar. In *Proc. ACL-02*.

Julia Hockenmaier and Mark Steedman. 2007. CCGbank: A Corpus of CCG Derivations and Dependency Structures Extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396.

Deirdre Hogan, Jennifer Foster, Joachim Wagner, and Josef van Genabith. 2008. Parser-based retraining for domain adaptation of probabilistic generators. In *Proc. INLG-08*.

Lidija Iordanskaja, Richard Kittredge, and Alain Polgúere. 1991. Lexical selection and paraphrase in a meaning-text generation model. In Cécile L. Paris, William R. Swartout, and William C. Mann, editors, *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, pages 293–312. Kluwer.

Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic*. Kluwer.

David Kauchak and Regina Barzilay. 2006. Paraphrasing for automatic evaluation. In *Proceedings of HLT-NAACL*.

Martin Kay. 1996. Chart generation. In *Proc. ACL-96*.

Irene Langkilde and Kevin Knight. 1998. The practical value of n-grams in generation. In *Proc. INLG-98*.

Irene Langkilde-Geary. 2002. An empirical verification of coverage and correctness for a general-purpose sentence generator. In *Proc. INLG-02*.

Bill MacCartney, Michel Galley, and Christopher D. Manning. 2008. A phrase-based alignment model for natural language inference. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 802–811, Honolulu, Hawaii, October. Association for Computational Linguistics.

Nitin Madnani. 2010. *The Circle of Meaning: From Translation to Paraphrasing and Back*. Ph.D. thesis, Department of Computer Science, University of Maryland College Park.

Crystal Nakatsu and Michael White. 2006. Learning to say it well: Reranking realizations by predicted synthesis quality. In *Proc. COLING-ACL-06*.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 1(29):19–52.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Bo Pang, Kevin Knight, and Daniel Marcu. 2003. Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *Proc. HLT/NAACL*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. ACL-02*.

Brian Roark, Murat Saraclar, Michael Collins, and Mark Johnson. 2004. Discriminative language modeling with conditional random fields and the perceptron algorithm. In *Proc. ACL-04*.

Hadar Shemtov. 1997. *Ambiguity Management in Natural Language Generation*. Ph.D. thesis, Stanford University.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proc. of the Association for Machine Translation in the Americas (AMTA-06)*.

Matthew Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. 2010. TER-plus: Paraphrase, semantic, and alignment enhancements to translation edit rate. *Machine Translation*, 23:117–127.

M. Stede. 1999. *Lexical Semantics and Knowledge Representation in Multilingual Text Generation*. Kluwer Academic Publishers.

Mark Steedman. 2000. *The Syntactic Process*. MIT Press.

Erik Velldal, Stephan Oepen, and Dan Flickinger. 2004. Paraphrasing treebanks for stochastic realization ranking. In *Proceedings of the 3rd Workshop on Treebanks and Linguistic Theories*.

Michael White and Rajakrishnan Rajkumar. 2009. Perceptron reranking for CCG realization. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 410–419, Singapore, August. Association for Computational Linguistics.

Michael White. 2004. Reining in CCG Chart Realization. In *Proc. INLG-04*.

Michael White. 2006a. CCG chart realization from disjunctive logical forms. In *Proc. INLG-06*.

Michael White. 2006b. Efficient Realization of Coordinate Structures in Combinatory Categorial Grammar. *Research on Language & Computation*, 4(1):39–75.

Shiqi Zhao, Xiang Lan, Ting Liu, and Sheng Li. 2009. Application-driven statistical paraphrase generation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 834–842, Suntec, Singapore, August. Association for Computational Linguistics.